

An Integrated Environment to Visually Construct 3D Animations

Enrico Gobbetti and Jean-Francis Balaguer[†]

Center for Advanced Studies, Research and Development in Sardinia

ABSTRACT

In this paper, we present an expressive 3D animation environment that enables users to rapidly and visually prototype animated worlds with a fully 3D user-interface. A 3D device allows the specification of complex 3D motion, while virtual tools are visible mediators that live in the same 3D space as application objects and supply the interaction metaphors to control them. In our environment, there is no intrinsic difference between user-interface and application objects. Multi-way constraints provide the necessary tight coupling among components that makes it possible to seamlessly compose animated and interactive behaviors. By recording the effects of manipulations, all the expressive power of the 3D user interface is exploited to define animations. Effective editing of recorded manipulations is made possible by compacting all continuous parameter evolutions with an incremental data-reduction algorithm, designed to preserve both geometry and timing. The automatic generation of editable representations of interactive performances overcomes one of the major limitations of current performance animation systems. Novel interactive solutions to animation problems are made possible by the tight integration of all system components. In particular, animations can be synchronized by using constrained manipulation during playback. The accompanying video-tape illustrates our approach with interactive sequences showing the visual construction of 3D animated worlds. All the demonstrations in the video were recorded live and were not edited.

Keywords

3D Interaction, 3D Widgets, Virtual Tools, 3D Animation, Local Propagation Constraints, Data Reduction, Object-Oriented Graphics.

1. INTRODUCTION

Modern 3D graphics systems allow a rapidly growing user community to create and animate increasingly sophisticated worlds. Despite their inherent three-dimensionality, these systems are still largely controlled by 2D WIMP user-interfaces. The lack of correlation between manipulation and effect and the high cognitive distance from users to edited models are the major drawbacks of this solution [13]. The inadequacy of user-interfaces based on 2D input devices and mindsets becomes particularly evident in the realm of interactive 3D animation. In this case, the low-bandwidth communication between user-interface and application and the restrictions in interactive 3D motion specification capabilities make it extremely difficult to define animations with straight-ahead actions. This inability to interactively specify the animation timing is a major obstacle in all cases where the spontaneity of the animated object's behavior is important [21; 35; 4].

In this paper, we present an expressive 3D animation environment that enables users to rapidly and visually prototype animated worlds with a fully 3D user-interface. A 3D device allows the specification of complex 3D motion, while virtual tools supply the interaction metaphors to control application objects. In our environment, there is no intrinsic difference between user-interface and application objects. Multi-way constraints provide the necessary tight coupling among components that makes it possible to compose animated and interactive behaviors. By recording the effects of manipulations, all the expressive power of the 3D user interface is exploited to define animations. Effective editing of recorded manipulations is made possible by compacting all continuous parameter evolutions with our data-reduction algorithm, designed to preserve both geometry and timing. Novel interactive solutions to animation problems are made possible by the tight integration of all system components. In particular, animations can be synchronized using constrained manipulation during playback.

In the following sections, we present an overview of the system, we make comparisons with related work, and we conclude with a view of future directions. The accompanying video-tape illustrates our approach with interactive sequences showing the visual construction of 3D animated worlds. All demonstrations in the video were recorded live and were not edited.

2. SYSTEM OVERVIEW

2.1 Dynamic Model

Our animation environment is built on top of VB2 [17; 18], a graphics architecture based on objects and constraints. During interaction, the user is the source of a flow of information propagating from input device sensors to manipulated models.

[†] CRS4, Scientific Visualization Group, Via Nazario Sauro 10, 09123 Cagliari, Italy.

E-mail: {gobbetti|balaguer}@crs4.it

WWW: <http://www.crs4.it/~gobbetti>

<http://www.crs4.it/~balaguer>



VB2 applications are represented by a network of interrelated objects, and the maintenance of relationships is delegated to a constraint-based change propagation mechanism. Different primitive elements represent the various aspects of the system's state and behavior: *active variables* store the system's state, domain-independent *hierarchical constraints* [9] maintain multi-way relations between active variables, *daemons* provide support for discrete simulation tasks, and *indirect expressions* allow constraints and daemons to dynamically locate their variables. Constraints are maintained using an efficient local propagation algorithm based on *Skyblue* [27; 17; 18]. The solver is domain-independent and can maintain a hierarchy of multi-way, multi-output dataflow constraints. The fact that constraint solving consists in performing method selection on the basis of constraint priorities and graph structure, without considering the variables' values, allows an effective application of a lazy evaluation strategy [17; 18]. The main drawback of such a local propagation algorithm is the limitation to acyclic constraint graphs. However, as noted by Sannella et al. [28], cyclic constraint networks are seldom encountered in the construction of user interfaces, and limiting the constraint solver to graphs without cycles gives enough efficiency and flexibility to create highly responsive complex interactive systems. In VB2, the objects' internal constraint networks are designed so as to reduce the possibility of creating cyclic constraint graphs. Runtime introduction of a constraint that would create a cyclic graph causes an exception that can be handled to remove the offending constraint¹.

The state manager behavior and the constraint solving techniques are detailed in [17; 18].

2.2 Interaction

The system's desktop configuration uses keyboard commands to trigger mode changes and animation playback, a *Spaceball* for continuous specification of spatial transformations, and a mouse for picking. Both hands are thus used simultaneously to input information. LCD shutter glasses provide binocular perception of the synthetic world. Since our main research goal is to explore the potentialities of 3D interaction, we do not provide a two-dimensional graphical user interface. A 3D cursor, controlled by the *Spaceball*, is used to select and manipulate objects of the synthetic world.

Direct manipulation and virtual tools are the two techniques used to input information. Both techniques involve using mediator objects that transform the cursor's movements into modifications of manipulated objects. Virtual tools are visible first class objects that live in the same 3D space as application objects and offer the interaction metaphor to control them. Their visual appearance is determined by a modeling hierarchy, while their behavior is controlled by an internal constraint network [18].

As in the real world, users configure their workspaces by selecting tools, positioning and orienting them in space, and binding them to application objects. At the moment of binding, the tool decides whether to accept the connection by checking if the application object contains all the needed information and by verifying that the constraint graph obtained by connecting the tool to the model can be handled by the underlying solver (i.e. it is acyclic). The binding mechanism is defined in a declarative way by using indirect constraints [18].

¹VB2's current constraint solver [17; 28] is unable to find acyclic solutions of potentially cyclic constraint graphs. An algorithm that removes this limitation is presented in [36].

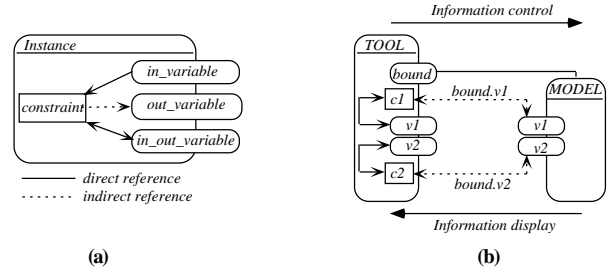


Figure 1a. Design notation

Figure 1b. Model and virtual tool

When bound, the tool changes its visual appearance to a shape that provides information about its behavior and offers semantic feedback. During manipulation, the tool's and the application object's constraint networks remain continuously connected, so as to ensure information propagation. Multiple tools can be active simultaneously in the same 3D environment in order to control all its aspects. The environment's consistency is continuously ensured by the underlying constraint solver. The bi-directionality of the relationships between user-interface and application objects makes it possible to use virtual tools to interact with a dynamic environment, opening the door to the integration of animation and interaction techniques.

2.3 Animation

By recording the effects of manipulations, animations can be sketched. In order to be able to edit the captured performance, a compact representation of continuous parameter evolution must be obtained. This representation must not only precisely approximate the shape of the initial parameter curves but also their timing. The data reduction algorithm must therefore treat the geometry and time components simultaneously in order to avoid the introduction of errors that would be difficult to control. We have developed an algorithm that incrementally builds, from the input sequence, a parametric B-spline preserving value and time of each input sample within a given tolerance. It is an incremental version of the Lyche and Mørken algorithm [22] that works in parallel with the interactive specification by considering only a small portion of the input curve at any time. Latency time and memory requirements for handling each portion of the curve are constant. Data reduction may therefore be performed concurrently with interactive parameter input, and the responsiveness of the application can be ensured when handling animations defined by any number of samples. The algorithm is presented in detail in [2; 4]. This performance-based approach complements key-framing by providing the ability to create animations with straight-ahead actions. It provides complete control over the animation shape and timing, while key-framing offers control only at a limited number of points.

The mediation of virtual tools makes it possible to sketch the evolution of non-geometric attributes, while constrained or free motion can be specified with 3D devices. Since these devices offer continuous control of spatial transformations, subtle synchronizations between position and orientation components can be directly specified. In our environment, straight-ahead animations are defined by expressing the desire to record parameter evolution during interaction. This is done simply by pressing a different mouse button when starting an interaction task. A controller object is connected to each animatable model and is responsible for monitoring model state changes. While recording, all changes are handled by the controller to feed the animation tracks. Continuous tracks apply the data reduction

algorithm to the incoming information, while discrete tracks simply store a change value event. During playback, information propagates from the animation tracks through the controllers and down to the models. All connections are realized by bi-directional constraints. Since playback constraints are weaker than interaction constraints, the user can take control over animated models during playback. Animations involving synchronizations with the environment's evolution can thus be specified by interacting during playback [5].

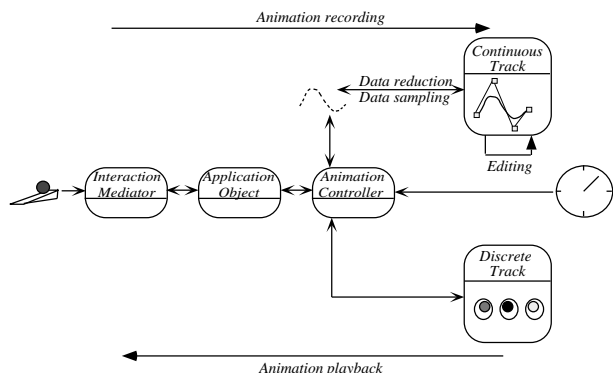


Figure 2. Interactive animation and playback

3. RELATED WORK

3.1 Constraint-based Architectures

Constraint-based architectures have long been used for 2D graphics systems (see [28] for a survey). In the 3D graphics world, one-way constraints are commonly employed to maintain dependencies between components [20; 34; 37; 38]. This type of constraint cannot easily model mutual relations between objects, thus hindering the tight coupling between user-interface and application objects [28]. Our system uses instead multi-way local propagation constraints, which offer support for two-way communication between objects while remaining efficient enough to ensure the responsiveness of the system [17; 18; 27]. *TBAG* [14] also uses multi-way constraints maintained by *Skyblue* [27], but its functional approach concentrates more on modeling time-varying behaviors than on creating interactive systems. Much effort has been spent in developing powerful numerical solvers for computer graphics (e.g. [7; 15; 16]). This work is complementary to ours, which focuses more on providing ways to interact with constrained environments. Such advanced solvers could replace local propagation in our system for the maintenance of numerical relationships.

3.2 Three-dimensional User Interfaces

Much recent research has focused on obtaining rich interaction with 3D environments by means of advanced devices and 3D interaction metaphors [8; 10; 11; 13; 16; 19; 26; 30; 32]. 3D widgets or manipulators, similar to our virtual tools, are presented in [13; 32]. These works focused on providing support for 3D widget construction, while we concentrate more on the integration of multiple tools in a single dynamic environment. We are not aware of any attempts to apply the results of 3D interaction research to enhance animation capabilities.

3.3 Performance Animation

A number of authors have proposed using live performances to drive computer animations (e.g. [1; 23; 33; 35]). We strive to bring the expressiveness of these approaches to general purpose animation systems running on graphics workstations. Instead of relying on advanced motion capture devices, we exploit our fully 3D user-interface to control the animated environment at a higher level of abstraction. The guiding approach proposed in [23] also seeks to provide better control of synthetic objects by raising the abstraction level of user interaction. That work concentrates on modeling complex behaviors in a discrete simulation framework, while we focus on providing intuitive user interfaces. A major limitation of current performance animation systems is the inability to build editable representations out of captured performances [35].

3.4 Data Reduction

Data reduction or curve fitting techniques have been successfully applied for the interactive specification of 2D or 3D curves or surfaces (e.g. [12; 24; 25; 29]). These techniques cannot be easily adapted to sketching animations of multi-dimensional parameters because they all exhibit one or more of the following problems: (i) restriction to 2D or 3D geometric constructions, (ii) lack of control on parameterization errors, and (iii) need to consider the entire input curve before reduction. An early attempt to use data-reduction for animation is described in [29]. In that system, path geometry and path timing specifications were decoupled, losing thus the advantages of performance approaches. Banks and Cohen [6] proposed for their drafting tool an incremental version of the Lyche and Mørken algorithm [22] that does not have the aforementioned drawbacks and could be used in a performance animation context. Their method shares with ours the idea of processing successive portions of the input curve which are then spliced together, but is unable to ensure constant latency times and memory needs [4].

4. CONCLUSIONS AND FUTURE WORK

In this video-paper, we have presented an integrated environment for the rapid and visual prototyping of 3D animated worlds. Using our fully 3D user-interface, non-professional users can swiftly create complex animations with pose-to-pose and straight-ahead techniques. Thanks to automatic data-reduction, animations created by interactive performances can then be effectively edited.

In our future work, we intend to develop new virtual tools and visualizations that will improve our 3D user interface for discrete and continuous track manipulation. To allow the system to adhere to timing requirements, we are developing time-critical techniques for controlling rendering complexity and constraint evaluation.

ACKNOWLEDGMENTS

The authors would like to thank Ronan Boulic for providing the walking engine used in the interactive sequences, Sally Kleinfeldt as well as Dean Allaman for helpful comments and suggestions, Angelo Mangili for technical help, and Michele Müller for doing the voice on the video.

This research was conducted by the authors while at the Swiss Federal Institute of Technology in Lausanne.

REFERENCES

- [1] Baecker RM (1969) Picture-driven Animation. *Proc. Spring Joint Computer Conference* 34: 273-288.
- [2] Balaguer JF (1993) *Virtual Studio: Un système d'animation en environnement virtuel*. PhD Thesis, Swiss Federal Institute of Technology in Lausanne.
- [3] Balaguer JF, Gobbetti E (1995) Animating Spaceland. To appear in *IEEE Computer Special Issue on Real-world Virtual Environments* 28(7).
- [4] Balaguer JF, Gobbetti E (1995) Sketching 3D Animations. To appear in *Proc. EUROGRAPHICS*.
- [5] Balaguer JF, Gobbetti E (1995) Supporting Interactive Animation using Multi-way Constraints. Submitted for publication.
- [6] Banks M, Cohen E (1990) Real-time Spline Curves from Interactively Sketched Data. *Proc. SIGGRAPH Symposium on Interactive 3D Graphics*: 99-107
- [7] Barzel R, Barr A (1988) A Modeling System Based on Dynamic Constraints. *Proc. SIGGRAPH*: 179-188.
- [8] Bier EA (1990) Snap-Dragging in Three Dimensions. *Proc. SIGGRAPH Symposium on Interactive 3D Graphics*: 193-204.
- [9] Borning A, Freeman-Benson B, Wilson M (1992) Constraint Hierarchies. *Lisp and Symbolic Computation* 5(3): 221-268.
- [10] Butterworth J, Davidson A, Hench S, Olano TM (1992) 3DM: A Three Dimensional Modeler Using a Head-Mounted Display. *Proc. SIGGRAPH Symposium on Interactive 3D Graphics*: 135-138.
- [11] Card SK, Robertson GG, Mackinlay JD (1991) The Information Visualizer: An Information Workspace. *Proc. SIGCHI*: 181-188.
- [12] Chou JJ, Piegl LA (1992) Data Reduction Using Cubic Rational Splines. *IEEE Computer Graphics and Applications* 12(3): 60-68.
- [13] Conner DB, Snibbe SS, Herndon KP, Robbins DC, Zeleznik RC, Van Dam A (1992) Three-Dimensional Widgets. *SIGGRAPH Symposium on Interactive 3D Graphics*: 183-188.
- [14] Elliott C, Schechter G, Yeung R, Abi-Ezzi S (1994) TBAG: A High Level Framework for Interactive, Animated 3D Graphics Applications. *Proc. SIGGRAPH*: 421-434.
- [15] Gleicher M (1993) A Graphics Toolkit Based on Differential Constraints. *Proc. UIST*: 109-120.
- [16] Gleicher M, Witkin A (1992) Through-the-Lens Camera Control. *Proc. SIGGRAPH*: 331-340.
- [17] Gobbetti E (1993) *Virtuality Builder II: Vers une architecture pour l'interaction avec des modes synthétiques*. PhD Thesis, Swiss Federal Institute of Technology in Lausanne.
- [18] Gobbetti E, Balaguer JF (1993) VB2: A Framework for Interaction in Synthetic Worlds. *Proc. UIST*: 167-178.
- [19] Herndon KP, van Dam A, Gleicher M (1994) Report: Workshop on the Challenges of 3D Interaction, CHI Bulletin, October.
- [20] Kass M (1992) CONDOR: Constraint-based Dataflow. *Proc. SIGGRAPH*: 321-330.
- [21] Lasseter J (1987) Principles of Traditional Animation Applied to 3D Computer Animation. *Proc. SIGGRAPH*: 35-44.
- [22] Lyche T, Mørken K (1987) Knot Removal for Parametric B-spline Curves and Surfaces. *Computer Aided Geometric Design* 4: 217-230.
- [23] McKenna M, Pieper S, Zeltzer D (1990) Control of a Virtual Actor: The Roach. *Proc. SIGGRAPH Symposium on Interactive 3D Graphics*: 165-174.
- [24] Plass M, Stone M (1983) Curve Fitting with Piecewise Parametric Cubics. *Proc. SIGGRAPH*: 229-239.
- [25] Pudet T (1994) Real Time Fitting of Hand Sketched Pressure Brushstrokes. *Proc. EUROGRAPHICS*: 205-220.
- [26] Sachs E, Roberts A, Stoops D (1990) 3-Draw: A Tool for Designing 3D Shapes. *IEEE Computer Graphics and Applications* 11(6): 18-26.
- [27] Sannella M (1994) Skyblue: A Multi-Way Local Propagation Constraint Solver for User Interface Construction. *Proc. UIST*: 137-146.
- [28] Sannella M, Maloney J, Freeman-Benson B, Borning A (1992) Multi-way versus One-way Constraints in User-Interfaces. *Software Practice and Experience* 23(5): 529-566.
- [29] Schneider PJ (1988) *Phoenix: An Interactive Curve Design System Based on the Automatic Fitting of Hand-Sketched Curves*. Master's Thesis, University of Washington.
- [30] Shaw C, Green M (1994) Two-Handed Polygonal Surface Design. *Proc. UIST*: 212-215.
- [31] Shelley KL, Greenberg DP (1982) Path Specification and Path Coherence. *Proc. SIGGRAPH*: 157-166.
- [32] Strauss PS, Carey R (1992) An Object-Oriented 3D Graphics Toolkit. *Proc. SIGGRAPH*: 341-347.
- [33] Tice S (1993) VActor Animation Creation System. *SIGGRAPH Tutorial 1*.
- [34] Upson C, Fulhauber T, Kamins D, Laidlaw D, Schlegel D, Vroom J, Gurwitz R, van Dam A (1989) The Application Visualization System: A Computational Environment for Scientific Visualization. *IEEE CG&A* 9(4): 30-42.
- [35] Walters G (1993) Performance Animation at PDI. *SIGGRAPH Tutorial 1*.
- [36] Vander Zanden B (1995) *An Incremental Algorithm for Satisfying Hierarchies of Multi-way, Dataflow Constraints*. Technical Report, University of Tennessee, Knoxville.
- [37] Zeleznik RC, Conner DB, Wlocka MM, Aliaga DG, Wang NT, Hubbard PM, Knepp B, Kaufman H, Hughes JF, van Dam A (1991) An Object-Oriented Framework for the Integration of Interactive Animation Techniques. *Proc. SIGGRAPH*: 105-112.
- [38] Zeltzer D, Pieper S, Sturman DJ (1989) An Integrated Graphical Simulation Platform. *Proc. Graphics Interface*: 266-274.